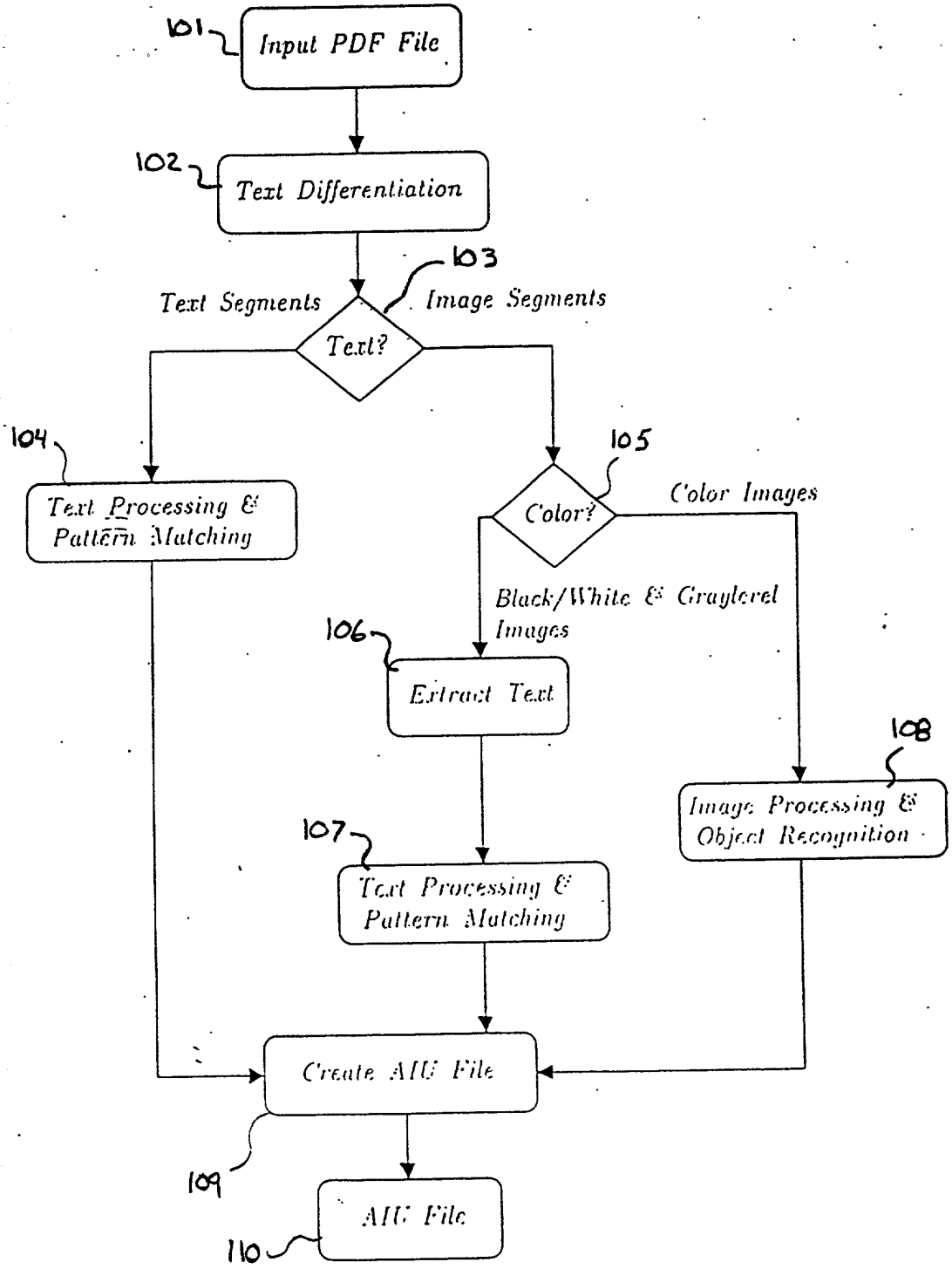
101
Input PDF File
102
Text Differentiation
103
Text Segments
Image Segments
Text?
104
Text Processing & Pattern Matching
105
Color?
Color Images
Black/White & Graylevel Images
106
Extract Text
107
Text Processing & Pattern Matching
108
Image Processing & Object Recognition
Create AIU File
109
AIU File
110

Fig. 1

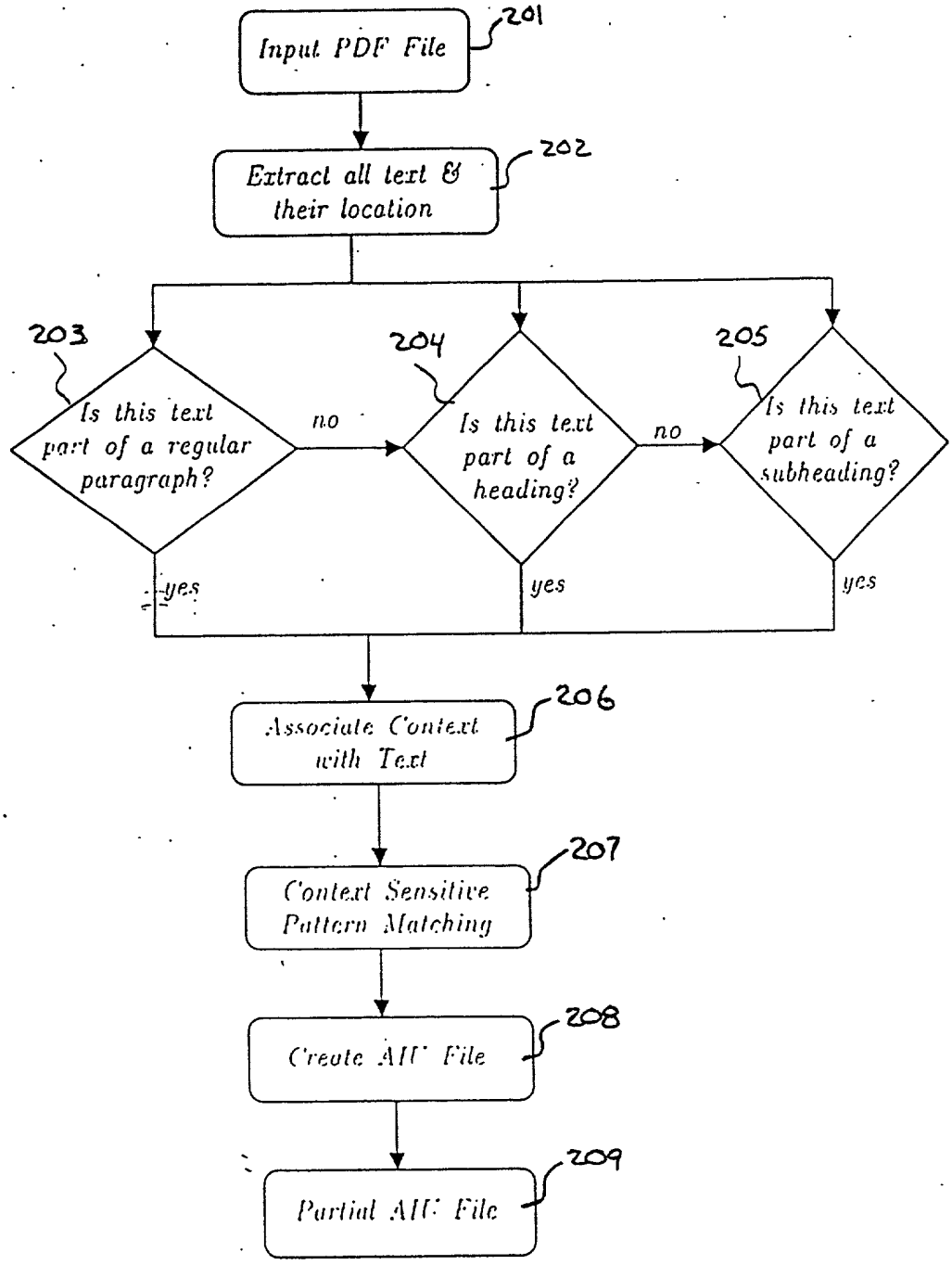
Input PDF File
201
Extract all text & their location
202
203
Is this text part of a regular paragraph?
no
204
Is this text part of a heading?
no
205
Is this text part of a subheading?
yes
yes
yes
Associate Context with Text
206
Context Sensitive Pattern Matching
207
Create AIF File
208
Partial AIF File
209

Fig. 2

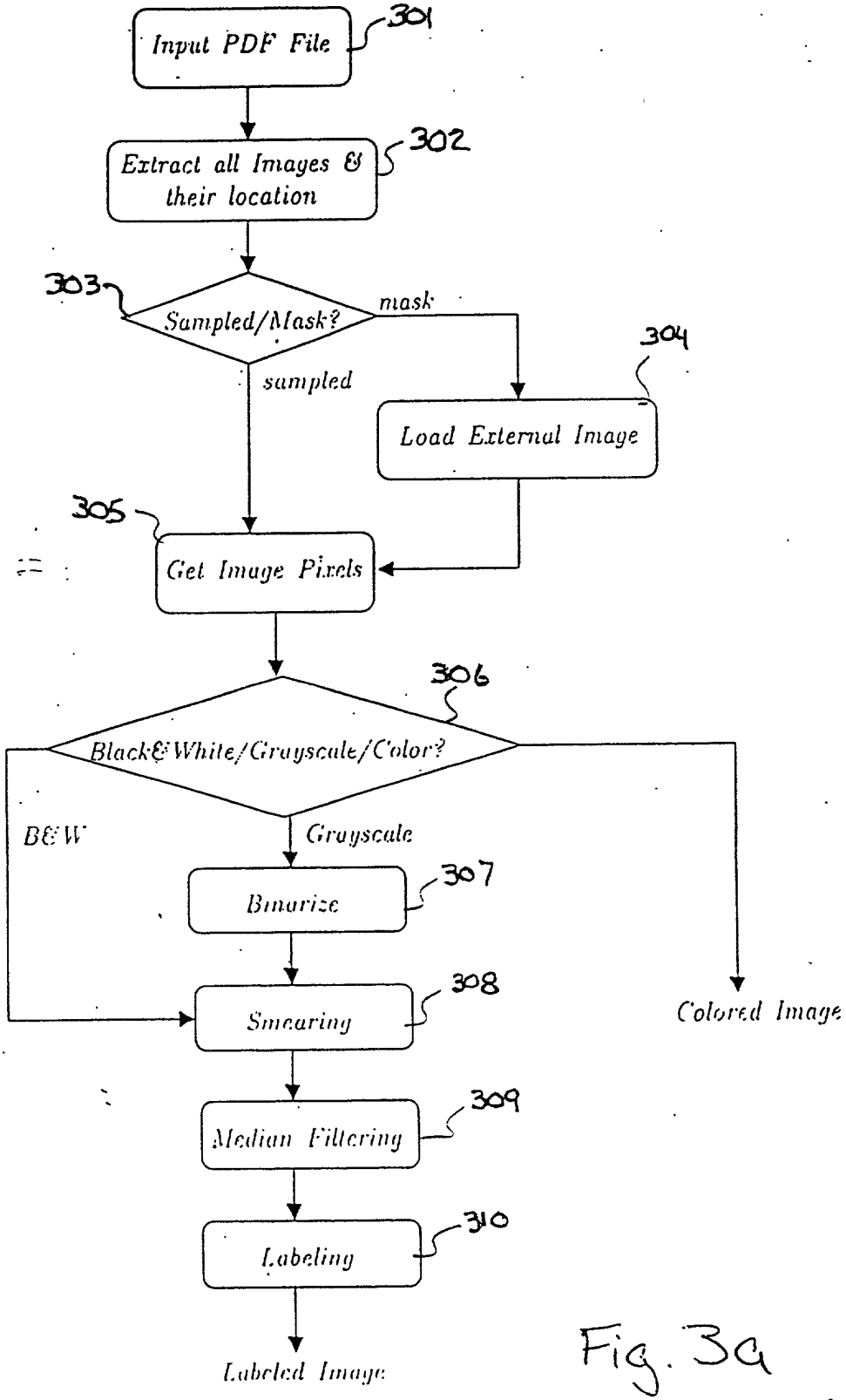
Input PDF File
301
Extract all Images & their location
302
303
Sampled/Mask?
mask
sampled
304
Load External Image
305
Get Image Pixels
306
Black&White/Grayscale/Color?
B&W
Grayscale
Colored Image
Binarize
307
Smearing
308
Median Filtering
309
Labeling
310
Labeled Image

Fig. 3a

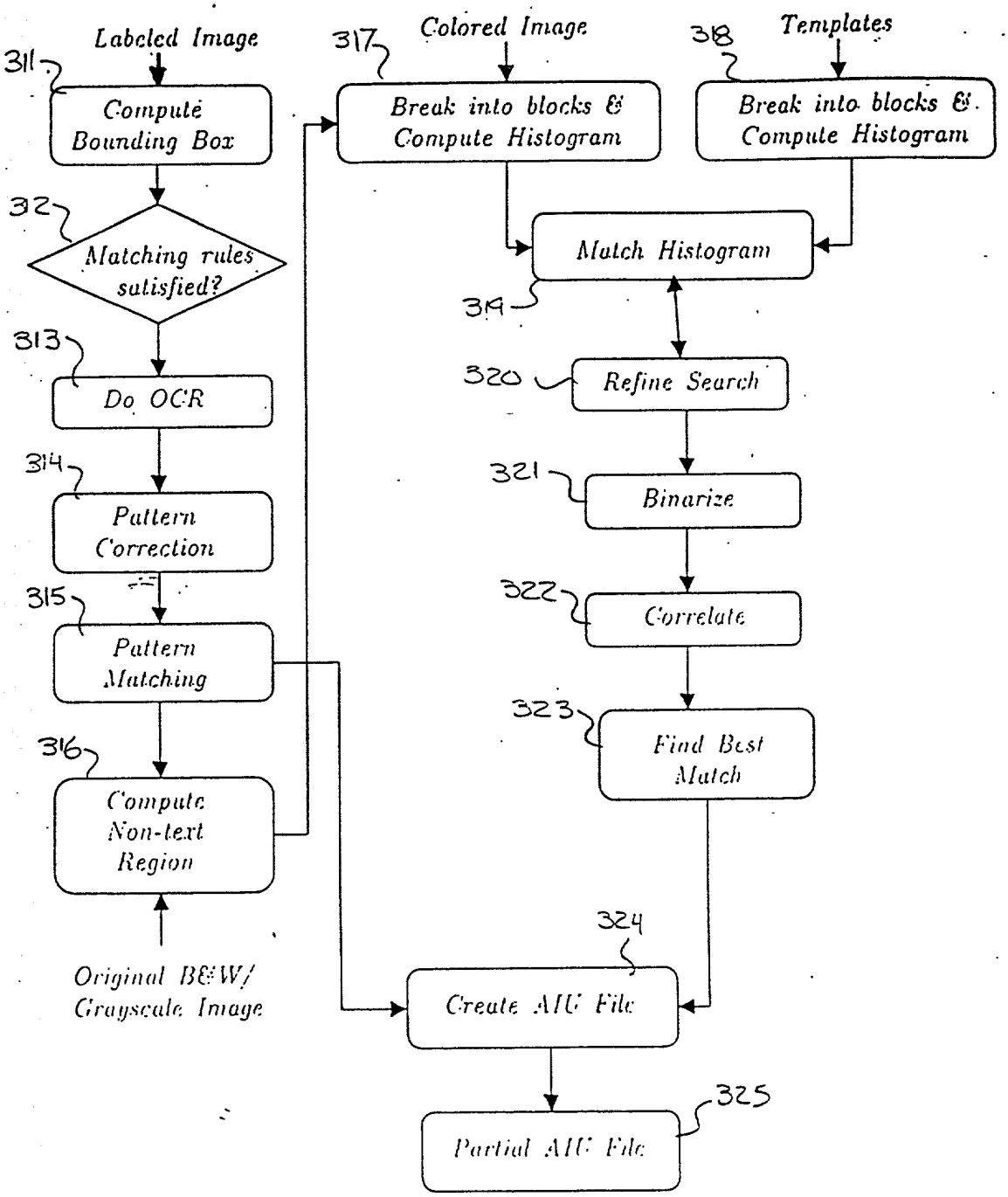
Labeled Image
311
Compute Bounding Box
32
Matching rules satisfied?
313
Do OCR
314
Pattern Correction
315
Pattern Matching
316
Compute Non-text Region
Original B&W/ Grayscale Image
317
Colored Image
Break into blocks & Compute Histogram
38
Templates
Break into blocks & Compute Histogram
Match Histogram
319
320
Refine Search
321
Binarize
322
Correlate
323
Find Best Match
324
Create AIC File
325
Partial AIC File

Fig. 3b

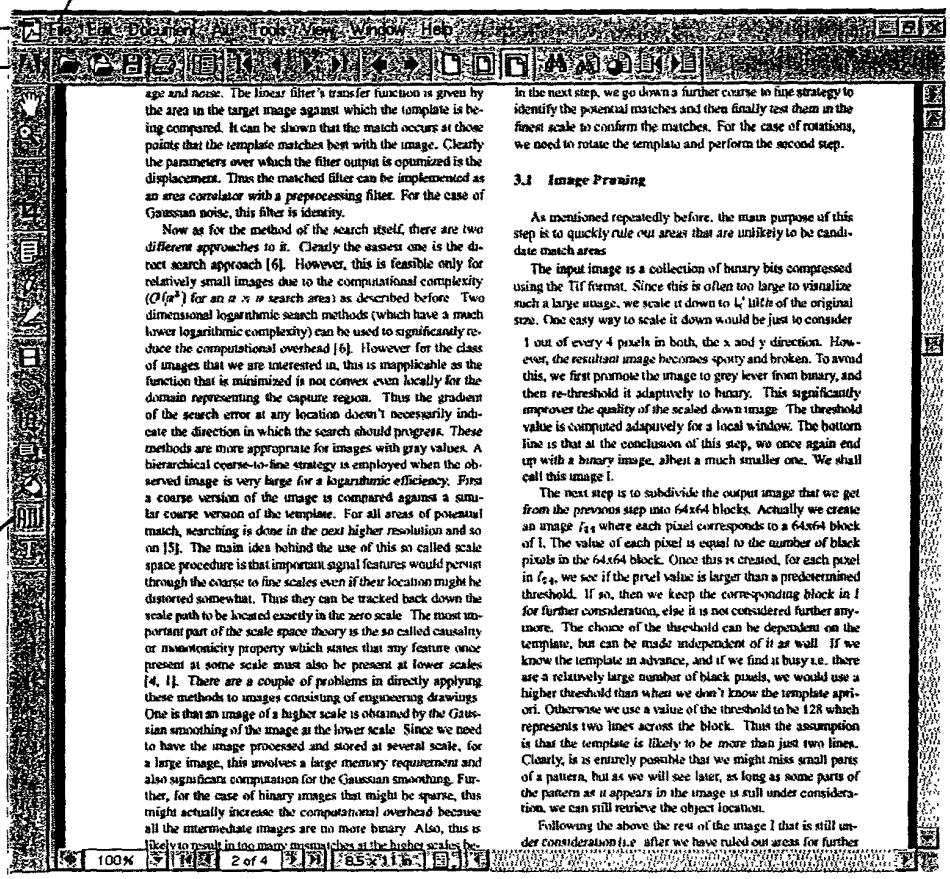
405
415
410
400
File Edit Document Arrange Tools View Window Help
age and noise. The linear filter's transfer function is given by the area in the target image against which the template is being compared. It can be shown that the match occurs at those points that the template matches best with the image. Clearly the parameters over which the filter output is optimized is the displacement. Thus the matched filter can be implemented as an area correlator with a preprocessing filter. For the case of Gaussian noise, this filter is identity.
Now as for the method of the search itself, there are two different approaches to it. Clearly the easiest one is the direct search approach [6]. However, this is feasible only for relatively small images due to the computational complexity ($O(n^2)$ for an $n \times n$ search area) as described before. Two dimensional logarithmic search methods (which have a much lower logarithmic complexity) can be used to significantly reduce the computational overhead [6]. However for the class of images that we are interested in, this is inapplicable as the function that is minimized is not convex even locally for the domain representing the capture region. Thus the gradient of the search error at any location doesn't necessarily indicate the direction in which the search should progress. These methods are more appropriate for images with gray values. A hierarchical coarse-to-fine strategy is employed when the observed image is very large for a logarithmic efficiency. First a coarse version of the image is compared against a similar coarse version of the template. For all areas of potential match, searching is done in the next higher resolution and so on [5]. The main idea behind the use of this so called scale space procedure is that important signal features would persist through the coarse to fine scales even if their location might be distorted somewhat. Thus they can be tracked back down the scale path to be located exactly in the zero scale. The most important part of the scale space theory is the so called causality or monotonicity property which states that any feature once present at some scale must also be present at lower scales [4, 1]. There are a couple of problems in directly applying these methods to images consisting of engineering drawings. One is that an image of a higher scale is obtained by the Gaussian smoothing of the image at the lower scale. Since we need to have the image processed and stored at several scale, for a large image, this involves a large memory requirement and also significant computation for the Gaussian smoothing. Further, for the case of binary images that might be sparse, this might actually increase the computational overhead because all the intermediate images are no more binary. Also, this is likely to result in too many mismatches at the higher scales, because in the next step, we go down a further course to fine strategy to identify the potential matches and then finally test them in the finest scale to confirm the matches. For the case of rotations, we need to rotate the template and perform the second step.
3.1 Image Pruning
As mentioned repeatedly before, the main purpose of this step is to quickly rule out areas that are unlikely to be candidate match areas.
The input image is a collection of binary bits compressed using the Tif format. Since this is often too large to visualize such a large image, we scale it down to $\frac{1}{4}$ width of the original size. One easy way to scale it down would be just to consider 1 out of every 4 pixels in both, the x and y direction. However, the resultant image becomes spotty and broken. To avoid this, we first promote the image to grey level from binary, and then re-threshold it adaptively to binary. This significantly improves the quality of the scaled down image. The threshold value is computed adaptively for a local window. The bottom line is that at the conclusion of this step, we once again end up with a binary image, albeit a much smaller one. We shall call this image I.
The next step is to subdivide the output image that we get from the previous step into 64x64 blocks. Actually we create an image I_{64} where each pixel corresponds to a 64x64 block of I. The value of each pixel is equal to the number of black pixels in the 64x64 block. Once this is created, for each pixel in I_{64}, we see if the pixel value is larger than a predetermined threshold. If so, then we keep the corresponding block in I for further consideration, else it is not considered further anymore. The choice of the threshold can be dependent on the template, but can be made independent of it as well. If we know the template in advance, and if we find it busy i.e. there are a relatively large number of black pixels, we would use a higher threshold than when we don't know the template apriori. Otherwise we use a value of the threshold to be 128 which represents two lines across the block. Thus the assumption is that the template is likely to be more than just two lines. Clearly, is is entirely possible that we might miss small parts of a pattern, but as we will see later, as long as some parts of the pattern as it appears in the image is still under consideration, we can still retrieve the object location.
Following the above the rest of the image I that is still under consideration i.e. after we have ruled out areas for further
100% 2 of 4 8.5 x 11 in

FIGURE 4